

eRide Serial Interface Protocol (eSIP)

User's Manual - Opus III ezRide



eRide Serial Interface Protocol (eSIP)

User's Manual - OPUS III ezRide

eTM000001
May 14, 2007
Revision 1

Content List

1	Purpose	4
2	Physical Interface	4
3	NMEA Sentence Format	4
4	Standard NMEA Outputs	4
4.1	GGA – Global Positioning System Fix Data	5
4.2	GLL – Geographic Position - Latitude/Longitude	6
4.3	GSA – GPS DOP and Active Satellites	6
4.4	GSV – Satellites in View	7
4.5	RMC – Recommended Minimum Navigation Information	7
4.6	ZDA – Time & Date	8
5	Standard NMEA Inputs	8
5.1	GPQ – NMEA Sentence Query	8
6	Proprietary NMEA Inputs	9
6.1	API – eRide GPS Core Library Interface	9
6.1.1	LATPROP – Enable Latency Position Propagation	9
6.1.2	OUTPROP – Enable Position Outage Propagation	9
6.1.3	STANDBY – Enter GPS Standby Mode	10
6.1.4	START – Start the GPS Core Library	10
6.1.5	STOP – Stop the GPS Core	11
6.2	CFG – System Configuration	12
6.2.1	FACTORYRESET – Return to Factory Settings	12
6.3	DBG – Control Debug Data Mode	12
6.3.1	DIAG – Enable Binary Debug Output	12
6.4	INFO – Get the PVT System Information	13
6.5	OUT – Configure the Standard NMEA Outputs	13
6.6	SYS – Control the PVT System	13
6.6.1	COMCTRL – Configure Serial Communications	13



eRide Serial Interface Protocol (eSIP)

User's Manual - OPUS III ezRide

eTM000001
May 14, 2007
Revision 1

7	Proprietary NMEA Outputs	14
7.1	ACK – Command Acknowledgement	14
7.2	ERR – System Error Report	15
7.3	INFO – Print System Information	15
7.4	RPx – Diagnostics Data Packet	16



eRide Serial Interface Protocol (eSIP)

User's Manual - OPUS III ezRide

eTM000001
May 14, 2007
Revision 1

1 Purpose

This document describes the eRide Serial communications Interface Protocol (eSIP) for controlling an eRide PVT module. The PVT module provides NMEA data over a serial port to another device. It also accepts input commands from a host. This document is intended to aid development of an external host to control the PVT module.

2 Physical Interface

The eRide PVT module uses a COM port interface model. Any physical support that uses this interface can be used, such as RS-232, USB or Bluetooth. The serial interface can operate at several different baud rates. The default configuration is 9600 baud, 8 data bits, no parity bit and one stop bit.

The serial interface is configured using the PERDSYS,COMCTRL command. The baud rate, parity and stop bits can be varied based on the host system (8 data bits are required). See the [COMCTRL – Configure Serial Communications](#) for more details.

3 NMEA Sentence Format

The interface is based on the NMEA sentence format. It's a well-defined format with many free viewers available on the internet. The first character is a dollar sign ('\$'). Data fields are separated by either a comma or a carat. The valid data character set is all characters from ASCII 0x22-0x7D except '!' (0x21), '\$' (0x24), '*' (0x2A) and '\' (0x5C). An eight-bit checksum, preceded by an asterisk (*), is included at the end of each line. Each sentence is terminated with a carriage return/line feed (CR/LF) sequence.

The NMEA standard specifies many standard output sentences. All standard sentences for Global Position System devices must start with \$GP. The standard also allows for vendor-specific data starting with \$P (proprietary) followed by a 3-letter designation for manufacturer (eRide has chosen ERD). All eRide proprietary sentences will begin with \$PERD.

4 Standard NMEA Outputs

The PVT application supports six standard NMEA output sentences (GGA, GLL, GSA, GSV, RMC and ZDA) per NMEA standard 0183 Version 3.01 (January 1, 2002). Each sentence will be output every second by default. The sentences can be independently enabled and disabled using the PERDOUT command described later in this document, as well as use differing transmission rates.

The NMEA sentence descriptions in this sentence are for reference only. The sentence formats are defined exclusively by the copyrighted document from NMEA.



4.1 GGA – Global Positioning System Fix Data

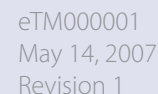
Time, Position and fix related data for a GPS receiver.

											11								
	1		2		3	4		5	6	7	8		9	10	11	12	13	14	15

```
$GPGGA,hhmmss.ss,llll.ll,a,yyyy.yy,a,x,xx,x.x,x.x,a,x.x,a,x.x,xxx*hh<CR><LF>
```

Field Number:

- 1) Universal Time Coordinated (UTC)
- 2) Latitude
- 3) N or S (North or South)
- 4) Longitude
- 5) E or W (East or West)
- 6) GPS Quality Indicator,
 - 0 - fix not available,
 - 1 - GPS fix,
 - 2 - Differential GPS fix
- 7) Number of satellites in view, 00 - 12
- 8) Horizontal Dilution of precision
- 9) Antenna Altitude above/below mean-sea-level (geoid)
- 10) Units of antenna altitude, meters
- 11) Geoidal separation, the difference between the WGS-84 earth ellipsoid and mean-sea-level (geoid), "-" means mean-sea-level below ellipsoid [not supported]
- 12) Units of geoidal separation, meters [not supported]
- 13) Age of differential GPS data, time in seconds since last SC104 type 1 or 9 update, null field when DGPS is not used [not supported]
- 14) Differential reference station ID, 0000-1023 [not supported]
- 15) Checksum





eRide Serial Interface Protocol (eSIP)

User's Manual - OPUS III ezRide

eTM000001
May 14, 2007
Revision 1

4.4 GSV – Satellites in View

	1	2	3	4	5	6	7		n

\$GPGSV,x,x,x,x,x,x,x,...*hh<CR><LF>

Field Number:

- 1) total number of messages (maximum of three)
- 2) message number
- 3) satellites in view (maximum of twelve)
- 4) satellite number
- 5) elevation in degrees
- 6) azimuth in degrees to true
- 7) SNR in dB

more satellite information like 4)-7) (up to four total SVs)

n) checksum

4.5 RMC – Recommended Minimum Navigation Information

	1		2	3		4	5		6	7	8	9		10		11	12	13

\$GPRMC,hhmmss.ss,A,1111.11,a,yyyy.yy,a,x.x,x.x,xxxx,x.x,a,a*hh<CR><LF>

Field Number:

- 1) UTC Time
- 2) Status, V = Navigation receiver warning
- 3) Latitude
- 4) N or S
- 5) Longitude
- 6) E or W
- 7) Speed over ground, knots
- 8) Track made good, degrees true
- 9) Date, ddmmyy
- 10) Magnetic Variation, degrees [not supported]
- 11) Magnetic Variation direction, E or W [not supported]
- 12) Mode Indicator

A – Autonomous, D – Differential, E – Estimated (dead reckoning),



M – Manual Input, S – Simulator, N – Data Invalid

13) Checksum

4.6 ZDA – Time & Date

UTC, day, month, year and local time zone

	1		2	3	4		5	6	7

\$GPZDA, hhmmss.ss, xx, xx, xxxx, xx, xx*hh<CR><LF>

Field Number:

- 1) UTC Time, e.g. 024611.08
- 2) UTC Day, (01 to 31)
- 3) UTC Month (01 to 12)
- 4) UTC Year (4 digits, i.e. 2007)
- 5) Local Time Zone Hours offset from UTC [not supported]
- 6) Local Time Zone Minutes offset from UTC [not supported]
- 7) Checksum

5 Standard NMEA Inputs

The PVT application supports the GPQ NMEA sentence query. This command allows a single NMEA sentence to be requested. The module will respond with a single transmission of the specified sentence.

5.1 GPQ – NMEA Sentence Query

- Format: \$GPGPQ,type*cks

type – Three-letter designation for sentence being requested. Valid values are GGA, GLL, GSA, GSV, RMC and ZDA

- Examples:

\$GPGPQ, GGA*3C	<i>Request the GGA sentence</i>
\$GPGPQ, GSV*3F	<i>Request the GSV sentence</i>



6 Proprietary NMEA Inputs

The PVT module will accept several proprietary commands from the host system. The NMEA input format will be used. The format requires an eight-bit checksum for each message, or the message will be ignored. An acknowledgement will be sent by the PVT module for each valid command received (see ACK – Command Acknowledgement for more details).

All input commands will start with the string \$PERD to indicate eRide-specific commands. This is required by the NMEA standard. \$P is used to indicate a proprietary (non-standard) NMEA. A 3-letter manufacturer designation then follows; eRide has selected ERD.

6.1 API –eRide GPS Core Library Interface

These commands control the GPS core algorithm. Each of these commands corresponds to an API call for configuring the eRide GPS library. The core library configuration command will be listed for each command. For more details on the API calls, please refer to *the eRide GPS Core Interface* document.

6.1.1 LATPROP – Enable Latency Position Propagation

This command controls the latency position propagation feature. The parameter is the number of milliseconds to propagate forward each calculated position fix. This feature is enabled by default with a setting of 0 milliseconds.

Core Library API function: *erSetLatencyPositionPropagation(int currentTimeOffset)*

o Format:

`$PERDAPI,LATPROP,time*cks`

time Number of milliseconds to propagate each calculated position fix. A time value of –1 will disable this feature.

o Examples:

<code>\$PERDAPI,LATPROP,0*2F</code>	Enable latency position propagation to latest time
<code>\$PERDAPI,LATPROP,500*2A</code>	Enable latency position propagation to 500 ms
<code>\$PERDAPI,LATPROP,-1*03</code>	Disable latency position propagation

6.1.2 OUTPROP – Enable Position Outage Propagation

This command controls the position outage propagation feature. The parameter is the number of seconds that the last calculated position is propagated forward using the last velocity in the event of a loss of satellite signals.



Core Library API function: `erSetPositionOutagePropagation(int maxSec)`

- o Format:

`$PERDAPI,OUTPROP,time*cks`

time Number of seconds to propagate the last valid position fix when the GPS signal is lost. A time value of 0 will disable this feature. The default value is ten seconds.

- o Examples:

`$PERDAPI,OUTPROP,5*3D`

Set outage position propagation to five seconds

`$PERDAPI,OUTPROP,0*38`

Disable outage position propagation

6.1.3 STANDBY – Enter GPS Standby Mode

This command puts the system into 27 MHz RTC standby mode. This stops the GPS algorithm from running, and it puts the chipset into a lower power mode. No parameters are required.

Core Library API function: `erGpsStandby(void)`

- o Format/Example:

`$PERDAPI,STANDBY*20`

Put GPS chipset into Standby mode

6.1.4 START – Start the GPS Core Library

This command starts the eRide GPS Core Library. The GPS core must be in Standby or Off state for this command to have any effect.

Core Library API functions: `erGpsStart(void)`, `erSetStartType(int startType)`

- o Format:

`$PERDAPI,START[,mode]*kk`

mode type of start to perform (default value is HOT). Acceptable values are shown in the table below. This parameter is optional. This parameter is used in the *erSetStartType* core API function.



eRide Serial Interface Protocol (eSIP)

User's Manual - OPUS III ezRide

eTM000001
May 14, 2007
Revision 1

Mode	Description
HOT	Hot start: uses previous position/time, requires local high accuracy satellite model.
WARM	Warm start: uses previous position/time
COLD	Cold start: uses local low accuracy satellite model
FACTORY	Factory start: uses default low accuracy satellite model stored in ROM
SIMCOLD	Simulator cold start: does not use any satellite models

o Examples:

\$PERDAPI, START*37	Do a hot start
\$PERDAPI, START, COLD, *IF	Do a cold start

6.1.5 STOP – Stop the GPS Core

This command stops the GPS core from running and puts the GPS chipset into the lowest power mode. No parameters are required.

Core Library API function: erGpsStop(void)

o Format/Example:

\$PERDAPI, STOP*6F	Put GPS chipset into Standby mode
--------------------	-----------------------------------

General Notes:

- After the STOP sentence is sent, the GPS algorithm and chipset are restarted by sending a PERDAPI,START command.
- The embedded controller remains active even though the algorithm is not longer executing.



6.2 CFG – System Configuration

6.2.1 FACTORYRESET – Return to Factory Settings

This command clears the non-volatile data. The data is stored in flash memory.

o Format/Example:

`$PERDCFG,FACTORYRESET*6C` *Clear all of the stored non-volatile data*

General Notes:

- This command should be executed when the system is "OFF" or in "STANDBY". It only clears the FLASH itself, not RAM copies of the FLASH data.
- If executed during execution, the effect of clearing previous run data may be lost. The data in RAM is normally stored to FLASH during shutdown, so while the FLASH will be cleared, the RAM data would just return the FLASH to the current settings when the shutdown command is executed. Other parameters are stored at various times during execution, so this may also nullify the effect of clearing FLASH.
- Clearing FLASH will require many CPU cycles. This may interfere with position fixing if the GPS algorithm is executing.
- This command CANNOT be undone. After a \$PERDCFG,FACTORYRESET command, the system is in "Factory Cold" mode with no previous run information or no RTC offset information. Behavior like TTFF will be affected. The system should normally be run with an outdoor antenna and allowed to fix for several minutes after executing this command to restore normal operating parameters.

6.3 DBG – Control Debug Data Mode

6.3.1 DIAG – Enable Binary Debug Output

This command controls the system debug output mode (see RPx - Diagnostics Data Packet). The only valid settings are ON and OFF; all other settings are ignored. The default value is OFF. This command must be sent with the baud rate set to 115.2 kbaud, or the command will be ignored. This command can be sent at any time, and the debug output will immediately begin. However, it is recommended that this command only be sent when the GPS core is stopped.

o Examples:

<code>\$PERDBG,DIAG,ON*48</code>	Enable diagnostics output
<code>\$PERDBG,DIAG,OFF*06</code>	Disable diagnostics output



6.4 INFO – Get the PVT System Information

Sending this command will return a string indicating the PVT information, including software revision number. It will be used to assist in problem reporting. The response is listed below in the Proprietary NMEA Outputs section.

- o Format/Example:

\$PERDINFO*0D

Request system information

6.5 OUT – Configure the Standard NMEA Outputs

These commands control the standard NMEA outputs.

- o Format:

\$PERDOUT, *type*, *interval**cks

type Three-letter designation for sentence being configured. Valid values are GGA, GLL, GSA, GSV, RMC and ZDA

interval Number of seconds to output each sentence. Zero seconds will disable the output. Negative values will be ignored.

- o Examples:

\$PERDOUT, GGA, 2*3E

Send the GGA sentence every two seconds

\$PERDOUT, GSV, 0*3F

Disable the GSV output sentence

6.6 SYS – Control the PVT System

These commands control the PVT module. Commands to start/stop the core and to put the circuit into low power mode are provided.

6.6.1 COMCTRL – Configure Serial Communications

This command configures the PVT serial communications port. The default settings are 9600 baud, eight data bits, no parity bit and one stop bit.



eRide Serial Interface Protocol (eSIP)

User's Manual - OPUS III ezRide

eTM000001
May 14, 2007
Revision 1

- o Format:

`$PERDSYS,COMCTRL,baud,parity,stopbits*cks`

baud Baud rate. Acceptable values are 4800, 9600, and 115200.

parity The parity format. Acceptable values are NONE, EVEN and ODD.

stopbits Number of stop bits between each byte. Acceptable values are 1 and 2.

Note that 8 data bits will always be used. The PVT module sends an ACK to the host in response to this command back at the original baud rate before changing to the new communications settings.

- o Examples:

`$PERDSYS,COMCTRL,9600,NONE,1*26`

Set to 9600 baud, no parity, one stop bit

`$PERDSYS,COMCTRL,115200,ODD,1*6B`

115200 baud, odd parity, one stop bit

7 Proprietary NMEA Outputs

7.1 ACK – Command Acknowledgement

This string is sent in response to all inputs to confirm successful receipt by the client. It is up to the PC host to implement any error handling procedures.

- o Format:

`$PERDACK,command,sequence*cks`

command Echoes the initial field of the command received by the client.

sequence Counter that increments for each receipt of a successful command. Range is from 000 to 255; rollover automatically occurs. This value will be a negative number for errors. Note that the command must still pass checksum in order to return an error.

-1: Unknown or bad command

- o Examples:

`$PERDACK,PERDAPI,001*20`

Confirm receipt of command

`$PERDACK,PERDJUNK,-1*4F`

Signal unknown command error



7.2 ERR – System Error Report

This string will be sent when the PVT module detects an error condition.

o Format:

```
$PERDERR,errornum,string*cks
```

errornum Type of error being reported. The error codes are listed below.

- 1: Shutdown failure
- 2: GPS communication failure
- 3: Non-volatile memory access failure

string Error message – arbitrary text field giving error details

o Example:

```
$PERDERR,2,Shutting down core*53
```

7.3 INFO – Print System Information

This string will be sent whenever the PVT module has been reset. It will also be sent in response to the PERDINFO input command from the host.

o Format:

```
$PERDINFO,product,revision,gpsmode,clientID*cks
```

product ASCII string describing the product.

revision Passes the revision numbers for all system components. The revisions are all concatenated into a single field.

gpsmode Current GPS core library state
0: GPS is off
1: GPS is running
2: GPS is in RTC standby mode
3: GPS is in Time and Peripherals mode



clientID The client ID of the module. The full client ID is CLIENT_xxxxxxxx. However, the CLIENT_ portion is not transmitted in this sentence. The ID can have up to 9 digits.

o Examples:

```
$PERDINFO,OPUS3PVT,ENP301A713006_11631805_H101843457,1,3000*14
```

7.4 RPx – Diagnostics Data Packet

This is the diagnostic output data. This is used to gather the GPS diagnostic data for debugging complex issues. The serial data sent by the PVT should be saved to a file and sent to the sales representative. The output data will contain both standard NMEA output sentences and special debug sentences. The debug sentences begin with \$PERDRP. These sentences are only usable by eRide factory engineers. Stop the GPS core using either the STOP or STANDBY command.

1. Send either the STOP or STANDBY command at the current baud rate.
2. Use \$PERDSYS,COMCTRL to set the baud rate to 115.2 kbaud.
3. On the PC, switch baud rate to 115.2 kbaud.
4. Start logging serial communication data to a file.
5. Send \$PERDDBG,DIAG,ON*48.
6. Send the desired START command. This must be sent at 115.2 kbaud.
7. Run the program and accumulate data.
8. Send either the STOP or STANDBY command at 115.2 kbaud.
9. Stop logging the communications to a file.
10. Send \$PERDDBG,DIAG,OFF*06.
11. Use \$PERDSYS,COMCTRL to return the baud rate to the desired NMEA setting.
12. Repeat steps 4-8 as many times as desired.
13. Send the data files to eRide for analysis.